

**Proyecto/Guía docente de la asignatura**

<b>Asignatura</b>	PARADIGMAS DE PROGRAMACIÓN		
<b>Materia</b>	INFORMÁTICA		
<b>Módulo</b>			
<b>Titulación</b>	GRADO EN ESTADÍSTICA		
<b>Plan</b>	549	<b>Código</b>	47076
<b>Periodo de impartición</b>	2º CUATRIMESTRE	<b>Tipo/Carácter</b>	Formación Básica
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	1º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	César Vaca Rodríguez		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5620 E-MAIL: <a href="mailto:cvaca@infor.uva.es">cvaca@infor.uva.es</a>		
<b>Horario de tutorías</b>	Véase <a href="http://www.inf.uva.es">www.inf.uva.es</a> >> Alumno >> Apoyo >> Tutorías		
<b>Departamento</b>	INFORMÁTICA (ATC, CCIA y LSI)		



## 1. Situación / Sentido de la Asignatura

### 1.1 Contextualización

El objetivo principal de esta asignatura es, en su parte teórica, el proporcionar al alumno una visión general de los distintos paradigmas (imperativo, funcional, lógico) y técnicas (orientación a objetos, orientación a eventos, genericidad) de programación existentes. La parte práctica se basa en el aprendizaje del lenguaje Python, como ejemplo característico de lenguaje de scripting, discurrendo en paralelo la enseñanza teórica de las distintas técnicas y paradigmas de programación con su correspondiente implementación en Python.

### 1.2 Relación con otras materias

La asignatura introduce técnicas y conceptos que se ampliarán en otras asignaturas, particularmente **Estructuras de Datos y Algoritmos** (TADs, Tipado Algebraico)

### 1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado y conseguido las habilidades y destrezas establecidas en la guía docente de la asignatura de **Fundamentos de Programación**. También es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

## 2. Competencias

### 2.1 Generales

Código	Descripción
G1	Capacidad para la gestión de la información: Incluye la capacidad para la búsqueda, manejo y exposición de información relevante de diversas fuentes, así como el manejo de las herramientas TIC necesarias
G2	Capacidad para la abstracción y el razonamiento crítico: El modelado y análisis de datos de problemas reales exige una elevada capacidad de abstracción, y el razonamiento crítico es necesario para hacer interpretaciones y establecer conclusiones y soluciones con rigor científico.
G3	Capacidad para la puesta al día y el auto-aprendizaje: Incluye la capacidad para la búsqueda de recursos que permitan la solución de nuevos problemas o de nuevas técnicas, en un medio científico y tecnológico en continua evolución

### 2.2 Específicas

Código	Descripción
E1	Recogida y tratamiento de datos: Incluye la capacidad para decidir sobre el diseño del procedimiento de obtención de datos. Capacidad para la búsqueda de información de fuentes diversas y para la elaboración de cuestionarios. Capacidad para manejar bases de datos y para llevar a cabo el tratamiento de los mismos.
E2	Descripción y síntesis de datos: Esta competencia es la que permitirá describir numéricamente y mediante gráficos diferentes características de interés de variables e individuos de los datos objeto de estudio, localización, escala, diferentes tipos de asociación, outliers,... con empleo y adaptación de la herramienta informática apropiada o programación específica.
E5	Extracción de conclusiones: Incluye la capacidad para interpretar los resultados del ajuste de un modelo seleccionado en términos del problema objeto de estudio, viendo su utilidad y/o proponiendo la necesidad de otras orientaciones del estudio.
E6	Presentación y comunicación de resultados: Incluye la capacidad para presentar los resultados de los análisis realizados, junto a las posibles soluciones a los problemas planteados por los demandantes del estudio en contextos diversos
E7	Dominio de conceptos y herramientas informáticas a nivel de usuario avanzado: programación, sistemas operativos, algoritmos, computación, inteligencia artificial, aprendizaje automático, almacenes y minería de datos, etc.
I1	Capacidad de análisis y síntesis
I3	Capacidad de organización y planificación
I4	Conocimientos de informática relativos al ámbito de estudio
I5	Resolución de problemas
I6	Comunicación oral y escrita en lengua nativa
I7	Conocimiento de lenguas extranjeras
P2	Razonamiento crítico
P3	Habilidades en las relaciones interpersonales
P4	Compromiso ético
S1	Aprendizaje autónomo
S2	Adaptación a nuevas situaciones
S3	Motivación por el trabajo bien hecho
S4	Iniciativa y espíritu emprendedor
S5	Creatividad



### 3. Objetivos

Código	Descripción
O.1	Comprender los distintos modelos de computación y paradigmas de programación.
O.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
O.3	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.
O.4	Conocer y saber utilizar lenguajes de scripting.
O.5	Conocer y ser capaz de interpretar las estructuras de los lenguajes de programación orientados a objeto y el contenido semántico de sus construcciones.





#### 4. Contenidos y/o bloques temáticos

##### Bloque 1: Paradigma Imperativo

Carga de trabajo en créditos ECTS: **2.0**

##### a. Contextualización y justificación

En este bloque temático se proporcionará una visión general de las distintas técnicas y paradigmas de programación, los modelos de cómputo en que se basan y los distintos lenguajes de programación y su evolución histórica. También se examinará en detalle el paradigma imperativo y los paradigmas de programación estructurada, procedimental y modular, estrechamente asociados a él. Por último se contempla con cierto detalle las técnicas de tratamiento de excepciones existentes.

##### b. Objetivos de aprendizaje

Código	Descripción
CI8.1	Comprender los distintos modelos de computación y paradigmas de programación.
CI8.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
CI14.1	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.
CI8.3	Conocer y saber utilizar lenguajes de scripting.

##### c. Contenidos

###### PARTE TEÓRICA

###### Tema 1. Introducción

- Conceptos fundamentales
- Modelos de cómputo
- Lenguajes de programación
- Familias de lenguajes y evolución histórica

###### Tema 2. Paradigma Imperativo

- Programación estructurada, procedimental y modular
- Tratamiento de excepciones
- Continuations, Closures, Corutinas
- Sistemas de tipado
- Datos estructurados y referencias
- Fortaleza y seguridad en sistemas de tipado

###### PARTE PRÁCTICA

###### Tema 1. Introducción

###### Tema 2. Elementos básicos del lenguaje

###### Tema 3. Funciones

###### Tema 4. Datos Estructurados



**d. Métodos docentes**

Actividad	Metodología
Clase de teoría	Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	Clase magistral participativa Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	Casos prácticos guiados por el profesor

**e. Plan de trabajo**

Para este bloque se estiman 20 horas presenciales distribuidas en 8 horas teóricas, 8 prácticas, 2 seminarios y 2 para efectuar las evaluaciones. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

**f. Evaluación**

La evaluación teórica de este bloque se efectuará mediante una prueba escrita de 2 horas de duración y se realizará en la séptima semana o en la anterior o posterior.

La evaluación práctica consistirá en la entrega y defensa de un trabajo práctico (2 horas de duración) y se realizará en la novena semana o en la anterior o posterior.

**g1. Bibliografía básica**

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

**g2. Bibliografía complementaria**

[Llamas] Llamas, C. "Introducción a la Informática. Modelos de Cómputo", Thomson, 2004.

**g3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)**

Consultar Aula Virtual de la asignatura

**h. Recursos necesarios**

Material de apoyo en la web:

[Transparencias] <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>

**i. Temporalización**

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas





## Bloque 2: Orientación a Objetos y Eventos

Carga de trabajo en créditos ECTS: **2.0**

### a. Contextualización y justificación

En este bloque temático se proporcionará una introducción a las técnicas de programación orientada a objetos, programación orientada a eventos y genericidad. Se intentará proporcionar una visión general que resalte las interacciones existentes entre estas técnicas.

### b. Objetivos de aprendizaje

Código	Descripción
CI8.5	Conocer y ser capaz de interpretar las estructuras de los lenguajes de programación orientados a objeto y el contenido semántico de sus construcciones.

### c. Contenidos

#### PARTE TEÓRICA

##### Tema 3. Orientación a Objetos

- Conceptos fundamentales
- Estructura estática: Clases, Encapsulamiento, Herencia
- Estructura dinámica: Objetos, tipos de métodos
- Polimorfismo, Ligadura dinámica
- Representación y gestión de memoria en O.O.

##### Tema 4: Orientación a Eventos

- Arquitectura de un sistema orientado a eventos
- Interfaces Gráficas de Usuario
- Paso de mensajes y manejadores de eventos
- Técnicas: Callbacks, Orientación a Objetos

##### Tema 5. Genericidad

- Objetivos. Técnicas principales.
- Genericidad y O.O.: Clases parametrizadas

#### PARTE PRÁCTICA

##### Tema 5. Orientación a Objetos

##### Tema 6. Programación en GUI



**d. Métodos docentes**

Actividad	Metodología
Clase de teoría	Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	Clase magistral participativa Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	Casos prácticos guiados por el profesor

**e. Plan de trabajo**

Para este bloque se estiman 20 horas presenciales distribuidas en 8 horas teóricas, 8 horas prácticas, 2 seminarios y 2 para efectuar evaluaciones. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

**f. Evaluación**

La evaluación teórica de este bloque se efectuará mediante una prueba escrita de 2 horas de duración y se realizará en la onceava semana o en la anterior o posterior.

La evaluación práctica consistirá en la entrega y defensa de un trabajo práctico (2 horas de duración) y se realizará en la última semana o en la penúltima.

**g1. Bibliografía básica**

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

**g2. Bibliografía complementaria**

[Meyer] Meyer, B. "Object-Oriented Software Construction" (2ª ed.), Prentice-Hall, 1997.

**g3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)**

Consultar Aula Virtual de la asignatura

**h. Recursos necesarios**

Material de apoyo en la web:

[Transparencias] <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>

**i. Temporalización**

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas



**Bloque 3: Paradigma Funcional**Carga de trabajo en créditos ECTS: **2.0****a. Contextualización y justificación**

En este bloque temático se estudiará el paradigma funcional tomando como base el lenguaje de programación Haskell. Se estudiará el tipado algebraico como una alternativa a las técnicas de orientación a objetos y genericidad. Por último se presentará al alumno el esquema map/filter/reduce de resolución de problemas.

Por último se presentará al alumno el paradigma lógico y sus características tomando como base el lenguaje Prolog.

**b. Objetivos de aprendizaje**

Código	Descripción
CI8.1	Comprender los distintos modelos de computación y paradigmas de programación.
CI8.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
CI14.1	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.

**c. Contenidos****PARTE TEÓRICA****Tema 6. Paradigma funcional**

- Objetivos, conceptos y abstracciones fundamentales
- Funciones como elementos de primer orden
- Concordancia de patrones y Evaluación diferida
- Tipado algebraico y genericidad
- Proceso de Listas: Map/Filter/Folder (Reduce)

**PARTE PRÁCTICA****Tema 7. Elementos de programación funcional****d. Métodos docentes**

Actividad	Metodología
Clase de teoría	Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	Clase magistral participativa Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	Casos prácticos guiados por el profesor



### e. Plan de trabajo

---

Para este bloque se estiman 20 horas presenciales distribuidas en 10 horas teóricas, 8 horas prácticas y 2 de seminarios. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

### f. Evaluación

---

La evaluación teórica de este bloque se efectuará en el examen final de la asignatura.

### g1. Bibliografía básica

---

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

### g2. Bibliografía complementaria

---

[Bird] Bird, R. "Introduction to Functional Programming using Haskell", (2ª ed.), Prentice-Hall, 1998.

[Clocksin] Clocksin, W.F., Mellish, C.S., "Programming in Prolog", Springer-Verlag, 1994.

[Arenas] Arenas, A., "Lógica Formal para Informáticos", Ed. Díaz de Santos, Madrid, 1996.

### g3. Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

---

Consultar Aula Virtual de la asignatura

### h. Recursos necesarios

---

Material de apoyo en la web:

[Transparencias] <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>

[Haskell] <http://www.lcc.uma.es/~blas/pfHaskell/gentle>

[SICP] <http://mitpress.mit.edu/sicp/full-text/book/book.html>

### i. Temporalización

---

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas

**5. Métodos docentes y principios metodológicos**

Actividad	Metodología
Clase de teoría	Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	Clase magistral participativa Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	Estudio e implementación de casos prácticos

**6. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA <sup>(1)</sup>	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	28	Estudio y trabajo autónomo individual	70
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	20
Laboratorios (L)	22		
Prácticas externas, clínicas o de campo			
Seminarios (S)	6		
Tutorías grupales (TG)			
Evaluación	4		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>
TOTAL presencial + no presencial			<b>150</b>

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor para otro grupo presente en el aula.



## 7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen teórico bloque 1 (T1)	20 %	Aproximadamente 7ª Semana. Recuperable en el examen extraordinario.
Entrega primera práctica (P1)	15 %	Aproximadamente 9ª Semana. No recuperable.
Examen teórico bloque 2 (T2)	20 %	Aproximadamente 11ª Semana. Recuperable en el examen extraordinario
Entrega segunda práctica (P2)	15 %	Aproximadamente 15ª Semana. No recuperable.
Examen teórico bloque 3 (T3)	30 %	En la fecha del examen ordinario. Recuperable en el examen extraordinario

### Procedimientos y Sistemas de Evaluación

Los exámenes teóricos consistirán en preguntas tipo test, preguntas de respuesta corta y la resolución de problemas sencillos.

La parte práctica consistirá en la resolución de problemas mediante la obtención del código Python adecuado. Además de la corrección del código, se valorará la utilización de las técnicas contempladas en la asignatura.

### CRITERIOS DE CALIFICACIÓN

**Convocatoria ordinaria:**

- o Se realizará mediante evaluación continua mediante las pruebas mencionadas en el apartado anterior. No se establece nota mínima en las pruebas para su contabilización.
- o La nota obtenida consistirá en la suma ponderada de las calificaciones obtenidas en los exámenes teóricos parciales y las prácticas.
- o Se considerará aprobado si  $T1+P1+T2+P2+T3 \geq 5.0$

**Convocatoria extraordinaria:**

- o Consistirá en un único examen en el que se podrá reevaluar cualquier combinación de los 3 bloques teóricos de la asignatura, a elección del alumno.
- o La parte práctica no se reevaluará.
- o Se considerará aprobado si  $T1+P1+T2+P2+T3 \geq 5.0$

## 8. Consideraciones finales

Las semanas indicadas para las evaluaciones son orientativas, pudiendo sufrir cambios (semana anterior o siguiente) por mejor acomodación del calendario.

En la página web principal de la asignatura se indicará el cronograma de actividades de la asignatura con sus fechas definitivas.

Página web de la asignatura: <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>